mod_limiter

Quick Start

This Apache httpd module provides the capability to restrict the number of concurrent connections clients may establish. This can be done by individual IP address, or range of addresses.

You can download the module by cloning the repository using the following command:

```
    > git clone https://git.earthdata.nasa.gov/scm/hdd/mod_limiter.git mod_limiter
```

or you may browse the repository on-line at:

https://git.earthdata.nasa.gov/projects/HDD/repos/mod_limiter/browse

Introduction

mod_limiter is a drop-in module for Apache httpd version 2.2 that can be used to provide connection concurrency limits on clients. It is configured using the standard Apache configuration directives.

Requirements

Please note the following requirements.

- mod_limiter has only been tested against Apache versions 2.2.22, 2.2.27, and 2.2.29, but should work with most 2.2 and 2.4 versions.
- *apxs* should be available. This is a command line tool used to build and install Apache modules, and may require Apache development libraries to be installed.

Installation

The mod_limiter source code is available from the ECC stash repository. It can be retrieved using git:

```
    > git clone https://git.earthdata.nasa.gov/scm/hdd/mod_limiter.git mod_limiter
```

This will create a subdirectory as follows:

```
mod_limiter ----- README
            |-- LICENSE
            |-- mod_limiter.h
            |-- mod_limiter.c
            |-- mod_limiter_net.c
            |-- mod_limiter_proc.c
```

```
        |-- mod_limiter_rule.c

        |-- mod_limiter_cfg.c
```

To build and install the module, go into the mod_limiter directory and use *apxs*:

```
> cd mod_limiter

> apxs -i -n mod_limiter -c mod_limiter.c mod_limiter_cfg.c mod_limiter_proc.c mod_limiter_net.c
mod_limiter_rule.c
```

In order for the new module to be loaded and used by Apache, you will need to restart httpd. However, it is a good idea to configure the module first.

Configuration

Configuration of Apache is generally done through httpd.conf, although there is a growing trend of different OS distributions to change that, so you may need to do a bit of searching to find the correct configuration file. The first step in configuring mod_limiter is to load it into Apache. This is done using the LoadModule directive (apxs may add this directive for you).

```
LoadModule limiter_module    modules/mod_limiter.so
```

This directive should go at the server root level. Once the module is loaded, you can use configuration directives specific to mod_limiter to configure the module.

Apache configuration directives generally fall into two main categories, *server level directives*, and *directory level directives*.

> **Server level directives** are configuration directives that are put at the server or virtual host level (i.e. not inside a directory/location/files configuration block).

> **Directory level directives** are those that go inside a directory/location/files configuration block.

mod_limiter only has server level configuration directives. These directives are:

> **LimiterEnable** Enables connection limiting. By default, connection limiting is disabled, and must be explicitly enabled. Valid values are `true/false/yes/no/1/0`.

> **ConnectionLimitRule** Defines a rule to limit connections from an IP address or range of IP addresses. Connection limit rules are described in more detail below.

Connection Limit Rules

A connection limit rule has the general form:

```
ConnectionLimitRule  <IP Address> < limit>
```

> **`<IP Address>`** is an IPV4 or IPV6 style IP address, and may include wildcards and/or ranges. The following are all valid values for `<IP Address>`.

```
192.168.1.34

192.168.1.*
```

```
192.168.1.[8-13]

192.168.[20-50].*

0:0:0:0:*:ffff:7f00:2

::[2-10]

any
```

The special case '**any**' means any IP address, but only if other rules do not apply. This is the equivalent of specifying every single IP address that does not match any other rule.

`<limit>` is a plain integer value, and gives the maximum number of concurrent connections from the IP addresses identified by `<IP Address>`. A value of 0 will prevent any connections from that IP address. Note that a value of 0 is not permitted with the '`any`' IP address.

If an IP address matches more than one limit rule, then ALL rules are applied and the connection will be restricted by the first failing rule (the one that reaches it's configured limit first). For example, a class C IP address may be limited to 5 connections, while a client with a specific IP address within that class C range may be limited to 3. The client can have at most 3 connections, but may be limited to fewer, depending upon how many other clients are connecting from the same class C IP address.

Example Configurations

Example 1

A simple configuration that explicitly limits a number of local network connections. All other clients will be restricted to at most 3 connections.

```
 <IfModule limiter_module>
    LimiterEnable             false

    ConnectionLimitRule         192.168.[50-50].* 5
    ConnectionLimitRule         ::[1-10]  4
    ConnectionLimitRule         127.0.0.1 8
    ConnectionLimitRule         0:0:0:0:0:ffff:7f00:2 6
    ConnectionLimitRule         any  3

</IfModule>
```

How It Works

The limiter module works by tracking the IP addresses of each client against each configured rule. If a client matches a rule, the count for that rule is incremented. When the client disconnects, the count for the matching rule is decremented. If a rule count reaches its configured limit, no more connections from clients that match that rule will be accepted (more specifically, they will be accepted, but immediately rejected).

The module stores the rule count information in shared memory (tuned for the specific MPM module being used). This means that there is some additional overhead when accepting connections - you will need to monitor this to make sure it does not adversely impact your operations.